REMARKS

Claims 1-4, 6-13, 15-21 and 23-37 are pending.

Applicants acknowledge and appreciate that the Examiner has withdrawn the rejection of claim 16 under 35 U.S.C. §101.

Claim Rejection – 35 U.S.C. §102(b)

Claims 1-4, 6-13, 15-21, 23-29 and 31-37 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,193,191 ***(McKeeman)***. Applicants respectfully traverse this rejection.

For ease of illustration, claim 1 is discussed first. Claim 1 calls for initiating compilation of a file in a processor-based system in advance of a request from a user to compile the file and detecting the user request to compile the file. Claim 1 also calls for indicating a status of the compilation of the file in response to detecting the user request. Initiating compilation of the file includes compiling the file in response to determining that the file has been modified.

The Examiner's rejection of claim 1 is improper because ***McKeeman***, as cited by the Examiner, fails to teach at least one of the claimed features. For example, claim 1 calls for initiating compilation of a file in a processor-based system <u>in advance of</u> a request from a user to compile the file. The Examiner argues that ***McKeeman*** teaches this feature because ***McKeeman*** teaches that recompilation "uses previously compiled code." *See* Final Office Action, p.3 (citing ***McKeeman***, col. 11, ll. 44-61). The passage cited by the Examiner, however, is completely silent with respect to initiating compilation of a file in a processor-based system <u>in advance of</u> a request from a user. Regardless of whether or not ***McKeeman*** teaches using previously compiled code for recompilation, ***McKeeman*** does <u>not</u> teach initiating compiling <u>in advance of</u> a request from a user, as called for in claim. In fact, ***McKeeman*** teaches that the recompilation is done <u>*after*</u> (not <u>in advance of</u>) a user initiates a compile. *See, e.g.,* ***McKeeman***, col. 5, ll. 8-18 (stating "[w]hen the developer has reached a point where he wishes to test the code he has written, *the compiler 11 is invoked*. The input to the compiler 11 is the source code text produced by the editor 10." (emphasis added)). Because ***McKeeman*** teaches compiling *after* a user request, this references does not, and cannot, teach the claimed feature of initiating compiling in advance of a request from a user, as called for in claim 1.

Claim 1 also calls for indicating a status of the compilation of the file in response to detecting the user request. The Examiner argues that ***McKeeman*** teaches this claimed feature. *See* Final Office Action, p.3 (citing ***McKeeman***, col. 5, ll. 21-13). The Examiner, as in the previous Office Action, improperly characterizes col. 5, lines 21-23 of ***McKeeman***. The Examiner states the cited passage of ***McKeeman*** teaches initiating compilation in response to determining that the file has been modified. This is incorrect. Rather, the cited passage teaches that, *after* the user initiates compilation (col. 5, lines 15-17), if a source text (module 12) has not been changed, then it is not recompiled.

> "When the developer has reached a point where he wishes to test the code he has written, the compiler 11 is invoked. The input to the compiler 11 is the source code text produced by the editor 10. There are typically a number of source code test buffers 12, one for each module of the application under development; according to one feature of the invention, those modules l2 which have not been changed or are not dependent upon changed code are not recompiled." ***McKeeman***, col. 5, ll. 15-23.

When properly read in context, the cited passage in ***McKeeman*** does not teach that the status of the compilation is indicated in response to detecting a user input, as argued by the Examiner. The Examiner's arguments cannot be correct because the compilation in ***McKeeman*** has not yet taken place when the user request is received. In other words, ***McKeeman*** fails to teach or suggest indicating the status of the compilation in response to detecting a user input. As such, ***McKeeman*** does not, and cannot, teach the claimed feature of the status of the compilation is indicated in response to detecting a user input, as called for in claim 1.

Claim 1 also calls for compiling the file in response to determining that the file has been modified. Again, the Examiner cites ***McKeeman*** , col. 5, ll. 21-23, as teaching this claimed feature. *See* Final Office Action, p.3. However, as Applicants have stated in the discussion above, ***McKeeman*** compiles when the user (developer) decides to compile, not in response to a file modification, as called for in claim 1.

For at least the aforementioned reasons, claim 1 and its dependent claims are allowable. For at least similar reasons, the remaining independent claims, and their respective dependent claims are also allowable.

Other claims are also allowable for additional features recited therein. For example, claim 24 calls for, *inter alia*, initiating processing of at least a portion of the modified source files (*i.e.*, one or more source files that have been modified) **before** receiving a request to process the modified files. The Examiner asserts that this feature is taught in ***McKeeman*** at column 11, lines 44-61. *See* Final Office Action, p.8. ***McKeeman*** fails to teach this feature. The cited passage describes reusing previously gathered information (such as compiled code) at recompilation if the source text has *not* changed. *See* ***McKeeman***, col. 11, ll. 44-61. Thus, this passage describes that, when recompilation is initiated, the compiler will compile only the changed files and will not recompile the unchanged source text, thereby saving unnecessary computation. This passage, however, does not describe initiating processing of one or more of the modified source files **before** a request to process the one or more source files (i.e., in ***McKeeman***'s case, a request to recompile the changed files) is received. The "recompilation" in ***McKeeman*** involves the reuse of previously compiled code derived from unchanged source text, and compiling only those source files that have been changed. Notably, the changed files in ***McKeeman*** are processed after the user initiates the request to recompile (col. 5, ll. 15-17). In contrast, claim 24 calls for initiating the processing of modified source files **before** receiving the request to process the modified source files. For at least this reason, claim 24 and its dependent claims are allowable.

The Examiner improperly characterizes column 11, lines 44-61 of ***McKeeman***. At page 8 of the Final Office Action, the Examiner argues that the cited passage of ***McKeeman*** teaches initiating processing of at least a portion of modified source files before receiving a request to process the modified files, and then receiving the request to process at least one of the modified files. This is incorrect. Rather, the cited passage teaches that processing of modified source files is initiated at recompilation time, i.e., after a request to process the modified files.

For at least these reasons, claim 24 and its dependent claims are allowable.

Turning to at least some of the dependent claims, claims 2-3, 12-13, 18-21, and 32-35 all recite the production of an object code file after compilation is initiated. In contrast, ***McKeeman*** is directed to generating debugged source code, not object code, and teaches later use of a different compiler to generate object code (col. 5, ll. 48-57).

Other dependent claims are also allowable for claimed features recited therein. For example, claims 29 and 33 recite the use of at least one marker or at least two markers, respectively, in identifying a section of the source file that should be compiled. ***McKeeman*** *fails* to teach markers, let alone the use of markers in identifying a section of the source file that should be compiled.

As such, Applicants request this rejection of claims 1-4, 6-13, 15-21, 23-29 and 31-37 under 35 U.S.C. §102(b) be withdrawn.

Claim Rejection – 35 U.S.C. §103

The Examiner, again, rejects claim 30 under 35 U.S.C. 103(a) as being unpatentable over ***McKeeman*** in view of U.S. Patent Publication No. 2005/0108682 ***(Piehler)***. Applicants respectfully traverse this rejection.

Claim 30 depends indirectly from independent claim 24. Because ***McKeeman*** doesn't disclose all of the features of claim 24 (for reasons discussed earlier), it likewise fails to teach the features of dependent claim 30. For at least this reason, claim 30 is allowable.

Arguments with respect to other dependent claims have been noted. However, in view of the aforementioned arguments, these arguments are moot and, therefore, not specifically addressed. To the extent that characterizations of the prior art references or Applicants' claimed subject matter are not specifically addressed, it is to be understood that Applicants do not acquiesce to such characterization.

In view of the foregoing, it is respectfully submitted that all pending claims are in condition for immediate allowance. The Examiner is invited to contact the undersigned attorney at (713) 934-4069 with any questions, comments or suggestions relating to the referenced patent application.

Respectfully submitted,

WILLIAMS, MORGAN & AMERSON, P.C.
CUSTOMER NO. 23720

Date: July 27, 2009

By: /Jaison C. John/
Jaison C. John, Reg. No. 50,737
10333 Richmond, Suite 1100
Houston, Texas 77042
(713) 934-4069
(713) 934-7011 (facsimile)
ATTORNEY FOR APPLICANT(S)